

AD-A212 514



Systems
Optimization
Laboratory

On the Identification of Local Minimizers in
Inertia-Controlling Methods for Quadratic
Programming

by
Anders L. Forsgren*, Philip E. Gill† and Walter Murray†

TECHNICAL REPORT SOL 89-11

July 1989

Department of Operations Research
Stanford University
Stanford, CA 94305

DTIC
ELECTE
SEP 18 1989
S B D
CB

STATEMENT A
This report contains information that is not to be released
without limitation.

4

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-4022

**On the Identification of Local Minimizers in
Inertia-Controlling Methods for Quadratic
Programming**

by
Anders L. Forsgren*, Philip E. Gill† and Walter Murray†

TECHNICAL REPORT SOL 89-11

July 1989



* Research supported by the Göran Gustafsson Foundation.

† Research and reproduction of this report were partially supported by the National Science Foundation Grant ECS-8715153; U.S. Department of Energy Grant DE-FG03-87ER25030, and Office of Naval Research Contract N00014-87-K-0142.

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do NOT necessarily reflect the views of the above sponsors.

Reproduction in whole or in part is permitted for any purposes of the United States Government. This document has been approved for public release and sale; its distribution is unlimited.

ON THE IDENTIFICATION OF LOCAL MINIMIZERS IN INERTIA-CONTROLLING METHODS FOR QUADRATIC PROGRAMMING

Anders L. FORSGREN*, Philip E. GILL† and Walter MURRAY‡

*Optimization and Systems Theory, Department of Mathematics
The Royal Institute of Technology, S-100 44 Stockholm, Sweden

†Department of Mathematics
University of California at San Diego, La Jolla, California 92093, USA

‡Systems Optimization Laboratory, Department of Operations Research
Stanford University, Stanford, California 94305-4022, USA

Technical Report SOL 89-11
July 1989

Abstract

The verification of a local minimizer of a general (i.e., nonconvex) quadratic program is in general an NP-hard problem. The difficulty concerns the optimality of certain points (which we call *dead points*) at which the first-order necessary conditions for optimality are satisfied, but strict complementarity does not hold.

One important class of methods for solving general quadratic programming problems are called *inertia-controlling quadratic programming (ICQP) methods*. We derive a computational scheme for proceeding at a dead point that is appropriate for a general ICQP method.

Keywords: Quadratic programming, local minimizer, NP-hardness.

1. Introduction

The *general quadratic programming problem* is to find a local minimizer of a quadratic function subject to linear constraints. In this paper, the problem considered is of the form

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & \varphi(x) = c^T x + \frac{1}{2} x^T H x \\ \text{subject to} & Ax \geq \beta, \end{array} \quad (1.1)$$

*Research partly supported by the Göran Gustafsson Foundation.

†‡ Research supported by the U.S. Department of Energy Grant DE-FG03-87ER25030; National Science Foundation Grant CCR-8413211; and the Office of Naval Research Contract N00014-87-K-0142.

where the *Hessian matrix* H is symmetric, and A is an $m_L \times n$ matrix. Of particular interest is the nonconvex case where H has an arbitrary distribution of positive, negative and zero eigenvalues.

Our attention will focus on the class of inertia-controlling methods for general quadratic programming. Inertia-controlling quadratic programming (ICQP) methods use a linearly independent subset of the constraints known as the *working set* to define a search direction and multiplier estimates. A unique feature of ICQP methods is that constraint deletions are restricted so as to control the inertia of the *reduced Hessian*, which is never permitted to have more than one nonpositive eigenvalue. Fletcher [Fle71] proposed the first ICQP method, and various methods within this class have also been proposed, see for example Gill *et al.* [GMSW84] and Gould [Gou86].

For any nonconvex quadratic program there may exist certain *dead points* at which all quadratic programming methods will find it difficult to proceed (see Section 2.6, for a precise definition of a dead point). The difficulty arises because the verification of such a point as a local minimizer of (1.1) is an NP-hard problem—see Murthy and Kabadi [MK87] and Pardalos and Schnitger [PS88]. Unfortunately, even if lower values of φ do exist in the neighborhood of a dead point, any number of constraints may need to be deleted simultaneously in order to compute a direction of improvement. Since existing ICQP methods can delete only one constraint at a time, they may be unable to proceed from a dead point.

The difficulties associated with existing ICQP methods at a dead point may be contrasted with the difficulties associated with the simplex method at a degenerate vertex. The simplex method is able to keep iterating at a degenerate vertex, but it may be necessary to perform a (possibly infinite) number of iterations during which the working set changes, but the vertex remains the same. By contrast, existing ICQP methods may terminate prematurely at a dead point which is not a local minimizer.

If progress is to be made at a dead point, a scheme must be devised for the identification and simultaneous deletion of more than one constraint from the working set. The computational and theoretical properties of such a scheme are presented in this paper. In Section 6 we show that at a dead point the proposed method behaves in a similar way to the simplex at a degenerate vertex—i.e., the algorithm is able to proceed, but there exists the danger of cycling. In order to guarantee finite termination, it is necessary to introduce procedures (such as a lexicographic ordering) similar to the anti-cycling procedures used in the simplex method.

In an ICQP method, the reduced Hessian is required to be positive definite at the starting point. In order to ensure this, it might be necessary to introduce *artificial constraints*. (The importance of such constraints is explained in Section 2.5.) Unfortunately, the presence of these constraints may introduce additional dead points which are not present in the original problem. However, we show that the computational scheme derived is able to treat the artificial-constraint case so that the artificial constraints cause no additional difficulties.

In order to describe the new scheme, we first review results on necessary and sufficient conditions for optimality in general quadratic programming. For a discus-

sion of these conditions, see for example Majthay [Maj71], Mangasarian [Man80], Contesse [Con80] or Borwein [Bor82]. The results presented here allow the presence of artificial constraints in the working set.

2. Basics

2.1. Notation

The following notation will be used throughout the paper:

The vector x denotes a feasible point of (1.1) to be examined.

The matrix A denotes the working-set matrix at x , and b denotes the associated right-hand side vector.

The scalar m denotes the number of rows in A .

The matrix Z denotes a matrix whose columns form a basis for the nullspace of A ; the reduced gradient and reduced Hessian of φ with respect to A are then $Z^T g(x)$ and $Z^T H Z$.

The vector g denotes the gradient of φ at x , i.e., $g = Hx + c$.

2.2. Terminology

The following terminology will be used:

A vector p is said to be a $\left\{ \begin{array}{ll} \text{descent direction} & \text{if } g^T p < 0. \\ \text{direction of positive curvature} & \text{if } p^T H p > 0. \\ \text{direction of negative curvature} & \text{if } p^T H p < 0. \\ \text{direction of zero curvature} & \text{if } p^T H p = 0. \\ \text{feasible direction} & \text{if } Ap \geq 0. \end{array} \right.$

A matrix D is said to be *copositive* if $v^T D v \geq 0$ for all $v \geq 0$.

A constraint $a_i^T x \geq \beta_i$ is said to be $\left\{ \begin{array}{ll} \text{active at } x & \text{if } a_i^T x = \beta_i. \\ \text{inactive at } x & \text{if } a_i^T x > \beta_i. \\ \text{violated at } x & \text{if } a_i^T x < \beta_i. \end{array} \right.$

2.3. Assumptions

The following assumptions are made:

- A1. The objective function, φ , is bounded from below in the feasible region.
- A2. All constraints active at x are in the working set.
- A3. The working-set matrix A has full row rank.



Session For	
GEI&I	<input checked="" type="checkbox"/>
Dist	<input type="checkbox"/>
Special	<input type="checkbox"/>
Availability (00-9)	
Dist	Special
A-1	

- A4. The point x satisfies the first-order necessary conditions for optimality, i.e., there exists a nonnegative Lagrange multiplier vector μ such that x and μ satisfy the Karush-Kuhn-Tucker equations

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ -\mu \end{pmatrix} = \begin{pmatrix} -c \\ b \end{pmatrix}. \quad (2.1)$$

- A5. The reduced Hessian, $Z^T H Z$, is positive definite.

2.4. The inertia of a matrix

Let M be any symmetric matrix. We denote by $i_p(M)$, $i_n(M)$ and $i_z(M)$ respectively the (nonnegative) numbers of positive, negative and zero eigenvalues of M . The *inertia* of M —denoted by $In(M)$ —is the associated integer triple (i_p, i_n, i_z) . The following lemma states an important relationship between the inertia of the *KKT-matrix*

$$K = \begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix}$$

and the reduced Hessian.

Lemma 2.1. *Given assumptions A3 and A5, the inertia of the KKT matrix K is $(n, m, 0)$.*

Proof. See Gould [Gou85, Lemma 3.4]. ■

Lemma 2.1 implies that K is nonsingular, so that the Lagrange multipliers in (2.1) are unique.

2.5. Inertia-controlling methods for quadratic programming

Associated with each iteration of an ICQP method is a linearly independent subset of the constraints known as the *working set*. The working set at the initial point x_0 must be chosen so that the reduced Hessian is positive definite. Thereafter, working set changes by only one constraint at each iteration and the reduced Hessian is never permitted to have more than one nonpositive eigenvalue.

ICQP methods depend critically on a procedure for finding an initial working set with an associated positive-definite reduced Hessian. In order to ensure that the reduced Hessian is positive definite, the initial working set may need to include “artificial” constraints that are not specified in the original problem. The only requirement for an artificial constraint is linear independence from constraints already in the working set. Artificial constraints do not restrict the feasible region, since the direction of the inequality need not be defined. As soon as an artificial constraint can be removed from the working set, it is eliminated from the problem. The strategy for choosing artificial constraints depends on the mechanics of the particular QP method and procedures for finding the initial working set are usually dependent on the method used to solve the KKT system. A simple example of a problem requiring

artificial constraints is given in Section 5. We emphasize that artificial constraints are not part of the original problem, but are an artefact of the solution method. The original constraints of the problem are referred to as *regular constraints*.

Once a constrained minimizer is found, an ICQP algorithm proceeds by deleting *one* constraint from the working set and finding either a feasible descent direction or a feasible direction of negative curvature. The constraint deletion is permitted only if the reduced Hessian is positive definite.

All ICQP methods generate search vectors and multipliers that satisfy the KKT equations. However, the equations may be solved either implicitly or explicitly, in which case one ICQP algorithm may appear to be very different from another. In this paper, only the form of equations to be solved is stated. For a discussion on the relationship between different ICQP methods, see Gill *et al.* [GMSW88].

Dead points with only regular constraints will be treated first. In Section 5, we consider the case when artificial constraints are present.

2.6. Dead points

If A contains only regular constraints, a *dead point* is defined to be a point satisfying assumptions A4 and A5 for which one or more components of the Lagrange multiplier vector μ are zero. We emphasize that such a point may not be a local minimizer.

Since a dead point satisfies the first-order necessary conditions for optimality, there exists no feasible descent direction. Therefore, it is necessary to find a feasible direction of negative curvature if an ICQP method is to proceed to find a local minimizer. Unfortunately, it may be impossible to compute a feasible direction of negative curvature by deleting only one constraint at a time, as can be seen from the following problem.

$$\begin{array}{ll} \underset{x \in \mathbb{R}^2}{\text{minimize}} & -x_1 x_2 \\ \text{subject to} & 0 \leq x_1 \leq 1 \\ & 0 \leq x_2 \leq 1. \end{array} \quad (2.2)$$

If the starting point is the origin, and both active constraints are in the working set, assumptions A1–A5 are satisfied. However, if either of the constraints is deleted from the working set, the resulting reduced Hessian is positive semidefinite and singular. Therefore, no feasible direction of negative curvature may be computed by deleting only one constraint. No more than one constraint may be deleted from the working set, since constraint deletion is permitted only when the reduced Hessian is positive definite. Therefore, an ICQP method must terminate at this point, although the origin is not a local minimizer for the QP.

In this situation—where neither a feasible descent direction nor a feasible direction of negative curvature may be found by deleting only one constraint—it is necessary to develop a scheme for deleting more than one constraint simultaneously if an ICQP method is to proceed.

2.7. Optimality conditions

In this section, necessary and sufficient conditions for x to be a local minimizer under assumptions A1–A5 are reviewed. It will be useful to distinguish between constraints with positive and zero multipliers. Without loss of generality we may assume that the rows of A are ordered such that

$$A = \begin{pmatrix} A_+ \\ A_0 \end{pmatrix},$$

where A_+ corresponds to rows with positive Lagrange multipliers and A_0 corresponds to rows with zero Lagrange multipliers. Let m_+ denote the number of rows in A_+ , and let m_0 denote the number of rows in A_0 . Also, let μ_+ denote the vector containing the m_+ positive components of μ .

The following two necessary and sufficient conditions for x being a local minimizer for (1.1) when assumption A2 holds are given by Majthay [Maj71] and Contesse [Con80].

C1. *The point x satisfies the first-order necessary conditions for optimality, i.e., there exists a nonnegative Lagrange multiplier vector μ , such that x and μ satisfy the KKT equations*

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ -\mu \end{pmatrix} = \begin{pmatrix} -c \\ b \end{pmatrix}.$$

C2. *It holds that $d^T H d \geq 0$ for all d such that $A_+ d = 0$ and $A_0 d \geq 0$.*

In his proof, Contesse derives an alternative formulation of Condition C2 involving the set of generators for the finite cone

$$\{p \mid A_+ p = 0, A_0 p \geq 0\}.$$

This formulation is described in Theorem 3.1 below. For the sake of completeness, Contesse's proof is reviewed with a notation relevant to our assumptions.

3. A Proof of the Optimality Conditions

Let Y_+ denote the $m_+ \times n$ matrix whose j -th column y_{+j} is defined to be the unique vector satisfying the equation

$$\begin{pmatrix} H & A_+^T & A_0^T \\ A_+ & 0 & 0 \\ A_0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_{+j} \\ -\rho_j \\ -\eta_j \end{pmatrix} = \begin{pmatrix} 0 \\ e_j \\ 0 \end{pmatrix}, \quad (3.1)$$

and let Y_0 denote the $m_0 \times n$ matrix whose j -th column y_{0j} is defined to be the unique vector such that

$$\begin{pmatrix} H & A_+^T & A_0^T \\ A_+ & 0 & 0 \\ A_0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_{0j} \\ -\lambda_j \\ -\theta_j \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ e_j \end{pmatrix}. \quad (3.2)$$

Equations (3.1) and (3.2) imply that the computation of y_{+j} and y_{0j} involves solving the KKT equations with a unit right-hand side. For a detailed discussion of the properties of the KKT equations in this context, see Gould [Gou86, Theorem 2.3].

Given Y_+ and Y_0 , let M denote the $n \times n$ matrix $M = \begin{pmatrix} Z & Y_+ & Y_0 \end{pmatrix}$.

Lemma 3.1. *The matrix M is nonsingular.*

Proof. It is enough to show that the columns of M are linearly independent. Assume that

$$Mv = Zv_z + Y_+v_+ + Y_0v_0 = 0.$$

Successive premultiplication of Mv by A_+ and A_0 gives $v_+ = 0$ and $v_0 = 0$. Since the columns of Z are independent, it follows that $v_z = 0$. ■

Lemma 3.2. *The sets*

$$\{p \mid Ap \geq 0\} \quad \text{and} \quad \{p \mid p = Zv_z + Y_+v_+ + Y_0v_0, v_+ \geq 0, v_0 \geq 0\}$$

are identical.

Proof. We may write p in the form $p = Zv_z + Y_+v_+ + Y_0v_0$, where v_+ and v_0 are nonnegative vectors. Premultiplication of p by A yields

$$Ap = \begin{pmatrix} A_+ \\ A_0 \end{pmatrix} p = \begin{pmatrix} v_+ \\ v_0 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

From Lemma 3.1 it follows that the columns of Z , Y_+ and Y_0 span \mathbb{R}^n . However, if any component of v_+ or v_0 is negative, the vector Ap will have at least one negative component. ■

Verification of the optimality of x is now equivalent to finding a local solution of the quadratic program

$$\begin{aligned} & \underset{p \in \mathbb{R}^n}{\text{minimize}} && g^T p + \frac{1}{2} p^T H p \\ & \text{subject to} && Ap \geq 0. \end{aligned} \tag{3.3}$$

Lemma 3.3. *The vector x is a local minimizer of (1.1) if and only if zero is a local minimizer of (3.3).*

Proof. The Taylor-series expansion of φ gives $g^T p + \frac{1}{2} p^T H p = \varphi(x+p) - \varphi(x)$. The vector Ap is nonnegative if and only if $A(x+p) \geq Ax$. Since every active constraint is included in A , the point x will not be a local minimizer of (1.1) if and only if there exists an infinite sequence $\{x^k\}_{k=1}^\infty$ converging to x such that $Ax^k \geq b$ and $\varphi(x^k) < \varphi(x)$. We need consider only those constraints in the working set because assumption A2 guarantees that if $\{x^k\}_{k=1}^\infty$ converges to x , all other constraints will

be satisfied for k sufficiently large. Similarly, the zero vector will not be a local minimizer of (3.3) if and only if there exists an infinite sequence $\{p^k\}_{k=1}^{\infty}$ converging to zero such that $Ap^k \geq 0$ and $g^T p^k + \frac{1}{2} p^{kT} H p^k < 0$. The proof is complete if we let $x^k = x + p^k$. ■

Lemma 3.4. *All elements of the matrices $Z^T H Y_+$ and $Z^T H Y_0$ are zero.*

Proof. Direct substitution in (3.1) yields $Z^T H y_{+j} = 0$ for $j = 1, \dots, m_+$ and direct substitution in (3.2) yields $Z^T H y_{0j} = 0$ for $j = 1, \dots, m_0$. ■

Lemmas 3.3 and 3.4 are now combined to show that the verification of x as a local minimizer is achieved by solving the QP problem

$$\begin{aligned} & \underset{v \in \mathbb{R}^n}{\text{minimize}} && \mu_+^T v_+ + \frac{1}{2} v_z^T Z^T H Z v_z + \frac{1}{2} v_+^T Y_+^T H Y_+ v_+ + v_+^T Y_+^T H Y_0 v_0 + \frac{1}{2} v_0^T Y_0^T H Y_0 v_0 \\ & \text{subject to} && v_+ \geq 0, \quad v_0 \geq 0. \end{aligned} \quad (3.4)$$

Lemma 3.5. *The vector x is a local minimizer of (1.1) if and only if zero is a local minimizer of (3.4).*

Proof. Problem (3.4) is derived from problem (3.3) by using the transformation

$$p = Mv = Zv_z + Y_+ v_+ + Y_0 v_0.$$

Assumption A4, equations (3.1) and (3.2), and Lemma 3.4 are used to simplify the objective function. The feasible region is obtained by using Lemma 3.2. Finally, Lemma 3.3 implies that zero is a local minimizer of (1.1) if and only if it is a local minimizer of (3.4). ■

Using these results it is possible to pose the problem of determining local optimality as a copositivity problem, as the following theorem shows.

Theorem 3.1. (Contesse [Con80]) *The point x is a local minimizer of (1.1) if and only if $Y_0^T H Y_0$ is copositive.*

Proof. Assume that $Y_0^T H Y_0$ is not copositive. Then there exists a nonnegative vector v_0 such that $v_0^T Y_0^T H Y_0 v_0$ is negative and zero is not a local minimizer of (3.4). Lemma 3.5 implies that x is not a local minimizer of (1.1).

Assume that $Y_0^T H Y_0$ is copositive. If zero is not a local minimizer of (3.4), there must exist an infinite sequence $\{v^k\}_{k=1}^{\infty}$ converging to zero such that

$$\mu_+^T v_+^k + \frac{1}{2} v_z^k T Z^T H Z v_z^k + \frac{1}{2} v_+^k T Y_+^T H Y_+ v_+^k + v_+^k T Y_+^T H Y_0 v_0^k + \frac{1}{2} v_0^k T Y_0^T H Y_0 v_0^k < 0,$$

where v_0^k and v_+^k are nonnegative. Since $Z^T H Z$ is positive definite and $Y_0^T H Y_0$ is copositive, it must hold that

$$\mu_+^T v_+^k + \frac{1}{2} v_+^k T Y_+^T H Y_+ v_+^k + v_+^k T Y_+^T H Y_0 v_0^k < 0.$$

At least one component of v_+^k must be positive, since the left-hand side is zero when v_+^k is zero. Since μ_+ is a positive vector, it must have a positive least component μ_{\min} , and we may write

$$\mu_{\min} e^T v_+^k + \frac{1}{2} v_+^k T Y_+^T H Y_+ v_+^k + v_+^k T Y_+^T H Y_0 v_0^k < 0,$$

where e is a suitably dimensioned vector with unit components. If both sides of this last equation are divided by the positive quantity $e^T v_+^k$, we obtain the inequality

$$\mu_{\min} + \frac{1}{2 e^T v_+^k} v_+^k T Y_+^T H Y_+ v_+^k + \frac{1}{e^T v_+^k} v_+^k T Y_+^T H Y_0 v_0^k < 0. \quad (3.5)$$

If we now consider this inequality as k goes to infinity, we note that μ_{\min} must be nonpositive, which contradicts the assumption that μ_+ is a positive vector. Hence, the zero vector is a local minimizer of (3.4) and Lemma 3.5 implies that x is a local minimizer of (1.1). ■

From this theorem, it follows that if we are able to check the $m_0 \times m_0$ matrix $Y_0^T H Y_0$ for copositivity, we are able to determine if x is a local minimizer.

4. On the Copositivity of a Matrix

It was shown in the previous section that the verification of optimality of a dead point x is equivalent to checking if the $m_0 \times m_0$ matrix $Y_0^T H Y_0$ is copositive. Once Y_0 is computed, the matrix $Y_0^T H Y_0$ may be calculated by performing direct matrix multiplications. However, the following lemma shows that the m_0 solutions of the equation (3.2) for $j = 1, \dots, m_0$ provide the matrix $Y_0^T H Y_0$.

Lemma 4.1. *If θ_j satisfies (3.2), then $Y_0^T H Y_0 e_j = \theta_j$ for $j = 1, \dots, m_0$.*

Proof. Direct substitution in (3.2) yields $y_{0j}^T H y_{0j} = e_j^T \theta_j$. ■

Copositive matrices have been studied extensively (see, e.g., Cottle *et al.* [CHL70] and Pereira [Per72]). The problem of deciding if a given matrix is copositive has been shown to be NP-hard, see Murty and Kabadi [MK87] and Pardalos and Schnitger [PS88]. Therefore, no computationally tractable method for solving the general problem is known.

However, there are special situations in which a matrix may be simply checked for copositivity. Two such situations are discussed in the following lemmas.

Lemma 4.2. *If the elements of the matrix $Y_0^T H Y_0$ are nonnegative, it is copositive.*

Proof. If $Y_0^T H Y_0$ is not copositive, there must exist a nonnegative vector v_0 such that $v_0^T Y_0^T H Y_0 v_0 < 0$. This is clearly impossible if all elements of $Y_0^T H Y_0$ are nonnegative. ■

Lemma 4.3. *If a diagonal element of $Y_0^T H Y_0$, say $y_{0i}^T H y_{0i}$ is negative, the matrix is not copositive. Moreover, the vector y_{0i} is a feasible direction of negative curvature.*

Proof. If $y_{0i}^T H y_{0i} < 0$, then clearly y_{0i} is a direction of negative curvature. Lemma 3.2 implies that y_{0i} is a feasible direction, as required. ■

It is also straightforward to check for copositivity when $Y_0^T H Y_0$ is a 2×2 matrix with nonnegative diagonal elements.

Lemma 4.4. *A 2×2 real symmetric matrix with nonnegative diagonal elements is not copositive if and only if its determinant is negative and its off-diagonal elements are negative. Moreover, if the matrix is not copositive, the eigenvector corresponding to the negative eigenvalue is a positive vector.*

Proof. See Cottle *et al.* [CHL70, Theorem 3.1]. ■

As a consequence of Lemma 4.4 the following lemma is immediate.

Lemma 4.5. *Assume that $Y_0^T H Y_0$ has nonnegative diagonal elements. Moreover, assume that it has a 2×2 principal submatrix with negative determinant and negative off-diagonal elements. A feasible direction of negative curvature for $Y_0^T H Y_0$ is given by the n_0 -vector whose nonzero elements are the components of the eigenvector corresponding to the negative eigenvalue of the 2×2 principal submatrix.*

Proof. It follows from Lemma 4.4 that it is possible to choose the eigenvector corresponding to the negative eigenvalue of this principal 2×2 submatrix with both components nonnegative. Hence, this eigenvector extended by zeros in the remaining $(n_0 - 2)$ positions is a feasible direction of negative curvature. ■

Now we propose a scheme for the verification of local optimality based on the lemmas above. First, it is shown that artificial constraints cause no additional difficulties.

5. Artificial Constraints in the Working Set

From the earlier discussion, it is clear that there may exist certain dead points at which the verification of local optimality is very difficult. In this section we demonstrate that this inherent difficulty need not be exacerbated by the imposition of artificial constraints.

To simplify the discussion, it will be necessary to distinguish between artificial and regular constraints. Accordingly, we partition A_0 and Y_0 such that

$$A_0 = \begin{pmatrix} A_R \\ A_A \end{pmatrix} \quad \text{and} \quad Y_0 = \begin{pmatrix} Y_R & Y_A \end{pmatrix},$$

where the subscript "R" denotes regular constraints and the subscript "A" denotes artificial constraints. Let m_R denote the number of rows of A_R and let m_A denote

the number of rows of A_A . Also let y_{Rj} denote the j -th column of Y_R and let y_{Aj} denote the j -th column of Y_A . When artificial constraints are present, the definition of a feasible direction will be changed as follows:

A vector p is said to be a *feasible direction* if $A_+p \geq 0$ and $A_Rp \geq 0$.

Note that the sign of the vector A_Ap is not restricted.

It is also necessary to use a slightly modified version of assumption 4:

A4'. The point x satisfies the first-order necessary conditions for optimality, i.e., there exists a Lagrange multiplier vector $\mu = (\mu_+ \ \mu_A \ \mu_R)^T$, with $\mu_+ \geq 0$, $\mu_R \geq 0$ and $\mu_A = 0$, such that x and μ satisfy the KKT equations

$$\begin{pmatrix} H & A_+^T & A_R^T & A_A^T \\ A_+ & 0 & 0 & 0 \\ A_R & 0 & 0 & 0 \\ A_A & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ -\mu_+ \\ -\mu_R \\ -\mu_A \end{pmatrix} = \begin{pmatrix} -c \\ b_+ \\ b_R \\ b_A \end{pmatrix}.$$

The difference between assumptions A4 and A4' is that the Lagrange multipliers of the artificial constraints are required to be zero. If an artificial constraint has a nonzero multiplier, it could be deleted from the working set to yield a feasible descent direction. Therefore, assumption A4' is appropriate for x being a constrained stationary point. Consequently, a point x is said to be a *dead point* if it satisfies assumptions A4' and A5.

Unfortunately, additional dead points may be added to the problem by imposing artificial constraints. Consider the problem

$$\begin{aligned} &\underset{x \in \mathbb{R}^2}{\text{minimize}} && -x_1x_2 \\ &\text{subject to} && -1 \leq x_1 \leq 1 \\ &&& -1 \leq x_2 \leq 1. \end{aligned} \tag{5.1}$$

If the starting point is the origin, no regular constraints are active and artificial constraints are needed to obtain a positive-definite reduced Hessian. If artificial bound constraints $x_1 = 0$ and $x_2 = 0$ are imposed, assumptions A1, A3, A4' and A5 are satisfied. However, as in problem (2.2), the origin is not a local minimizer and no feasible direction of negative curvature may be obtained by deleting only one artificial bound.

It might seem necessary that an arbitrary (unknown) number of artificial constraints must be deleted to give a feasible direction of negative curvature (if one exists). However, we shall show below that such a direction may be computed by making only *one or two* artificial constraints leave the working set.

In order to show this, we determine constraints in the working set that may be deleted yielding a positive-definite reduced Hessian. This may be determined from the solution of (3.1) and (3.2), as observed by Gould [Gou86, Theorem 2.3] and reviewed in the following lemma.

Lemma 5.1. *If a constraint corresponding to a positive diagonal element of $Y_0^T H Y_0$ is deleted from A , the resulting reduced Hessian remains positive definite.*

Proof. Let y_{0i} correspond to the deleted constraint $a_i^T x \geq \beta_i$. Lemma 3.1 implies that y_{0i} is independent of the columns of Z and it follows from (3.2) that $A y_{0i} = e_i$. Therefore, a basis for the new nullspace is obtained by adding the column y_{0i} to Z . Lemma 3.4 implies that $Z^T H y_{0i}$ is zero. Hence, the fact that $y_{0i}^T H y_{0i}$ is positive implies that the new reduced Hessian remains positive definite. ■

In order to distinguish between artificial and regular constraints we partition $Y_0^T H Y_0$ such that

$$Y_0^T H Y_0 = \begin{pmatrix} Y_R^T H Y_R & Y_R^T H Y_A \\ Y_A^T H Y_R & Y_A^T H Y_A \end{pmatrix}.$$

It follows from Lemma 4.3 that if a diagonal element of $Y_A^T H Y_A$ is negative, a feasible direction of negative curvature can be computed. By Lemma 5.1 it follows that if a diagonal element of $Y_A^T H Y_A$ is positive, the corresponding artificial constraint can be deleted and the new reduced Hessian will be positive definite. Clearly, unless all diagonal elements of $Y_A^T H Y_A$ are zero, either a feasible direction of negative curvature can be computed or an artificial constraint can be deleted.

Lemma 5.2. *If two diagonal elements of $Y_A^T H Y_A$, say $y_{Ai}^T H y_{Ai}$ and $y_{Aj}^T H y_{Aj}$ are zero, and $y_{Ai}^T H y_{Aj}$ is nonzero, the point x is not a local minimizer. Moreover, either $y_{Ai} - y_{Aj}$ or $y_{Ai} + y_{Aj}$ is a feasible direction of negative curvature.*

Proof. Direct calculation yields $(y_{Ai} + y_{Aj})^T H (y_{Ai} + y_{Aj}) = -(y_{Ai} - y_{Aj})^T H (y_{Ai} - y_{Aj}) = 2y_{Ai}^T H y_{Aj} \neq 0$. Hence, either $y_{Ai} + y_{Aj}$ or $y_{Ai} - y_{Aj}$ is a direction of negative curvature. Feasibility follows from the relations $A_+(y_{Ai} \pm y_{Aj}) = 0$ and $A_R(y_{Ai} \pm y_{Aj}) = 0$. ■

This lemma demonstrates that unless the matrix $Y_A^T H Y_A$ is zero, either a feasible direction of negative curvature can be computed or an artificial constraint can be deleted.

Lemma 5.3. *If the diagonals of $Y_A^T H Y_A$ are zero, and an element of $Y_R^T H Y_A$ (say $y_{Ri}^T H y_{Aj}$) is nonzero, the point x is not a local minimizer and a feasible direction of negative curvature may be computed.*

Proof. Let p be a vector of the form $\alpha_i y_{Ri} + \alpha_j y_{Aj}$. Direct calculation yields that p is feasible if α_i is nonnegative. The quantity $p^T H p$ may be expressed as

$$p^T H p = \begin{pmatrix} \alpha_i & \alpha_j \end{pmatrix} \begin{pmatrix} y_{Ri}^T H y_{Ri} & y_{Ri}^T H y_{Aj} \\ y_{Ri}^T H y_{Aj} & 0 \end{pmatrix} \begin{pmatrix} \alpha_i \\ \alpha_j \end{pmatrix}.$$

Consider the 2×2 matrix T given by

$$T = \begin{pmatrix} y_{Ri}^T H y_{Ri} & y_{Ri}^T H y_{Aj} \\ y_{Ri}^T H y_{Aj} & 0 \end{pmatrix}.$$

Since $y_{Ri}^T H y_{Aj}$ is nonzero, T has one negative and one positive eigenvalue. It has orthogonal eigenvectors, since it is a real symmetric matrix. Hence, α_i and α_j may be chosen so that p is the eigenvector corresponding to the negative eigenvalue, with α_i nonnegative. For those values of α_i and α_j , the vector p will be a feasible direction of negative curvature. ■

Clearly, whenever a component of $Y_R^T H Y_A$ is nonzero, either an artificial constraint can be deleted or a feasible direction of negative curvature can be computed. To summarize, the following result holds when artificial constraints are present in the working set.

Theorem 5.1. *If $Y_A^T H Y_A$ has nonpositive diagonal elements, then x is a local minimizer of (1.1) if and only if $Y_R^T H Y_R$ is copositive and $Y_R^T H Y_A$ and $Y_A^T H Y_A$ are zero.*

Proof. If $Y_R^T H Y_A$ or $Y_A^T H Y_A$ are nonzero, there exists a feasible direction of negative curvature and x cannot be a local minimizer.

Assume that $Y_R^T H Y_R$ is not copositive. In this case, a feasible direction of negative curvature may be computed and the local optimality of x is contradicted.

Assume that $Y_R^T H Y_R$ is copositive and $Y_R^T H Y_A$ and $Y_A^T H Y_A$ are zero. Using a similar analysis to that for the regular-constraint case, we can make the following assertions. As in Lemma 3.2, partition the vector v_0 such that

$$v_0 = \begin{pmatrix} v_R \\ v_A \end{pmatrix}$$

and replace the constraint $v_0 \geq 0$ in (3.4) by $v_R \geq 0$. If x is not a local minimizer of (1.1) there must exist an infinite sequence $\{v^k\}_{k=1}^{\infty}$ converging to zero such that

$$\mu_{\min} + \frac{1}{2e^T v_+^k} v_+^k T Y_+^T H Y_+ v_+^k + \frac{1}{e^T v_+^k} v_+^k T Y_+^T H Y_R v_R^k + \frac{1}{e^T v_+^k} v_+^k T Y_+^T H Y_A v_A^k < 0.$$

Again, if we consider this inequality as k goes to infinity, we obtain the required contradiction. ■

Consequently, if assumptions A1–A3, A4' and A5 hold, the artificial constraints will cause no extra problem in determining if x is a local minimizer. There remains the hard question of verifying that the matrix $Y_R^T H Y_R$ is copositive.

6. Computation of Directions of Negative Curvature

In this section, we propose an extension to ICQP methods that will allow progress to be made at a dead point. Algorithm 6.1 provides a means of computing a direction of negative curvature by making one or two active constraints inactive. Lemma 6.1 below indicates that the algorithm will terminate with either a direction of negative curvature or the conclusion that x is a local minimizer.

Lemma 6.1. *Algorithm 6.1 will terminate in at most m_0 steps. Moreover, if termination occurs without the computation of a direction of negative curvature, x is a local minimizer of (1.1).*

Proof. At each step, either the algorithm terminates or a constraint is deleted from the working set. Since there are only m_0 constraints to delete, the algorithm must stop in at most m_0 steps.

If $Y_A^T H Y_A$ has a positive diagonal element, the corresponding artificial constraint is deleted. Since this deletion will be repeated until every diagonal element of $Y_A^T H Y_A$ is nonpositive, we may assume that $Y_A^T H Y_A$ has nonpositive diagonal elements. At this point, if no direction of negative curvature is computed, the matrices $Y_R^T H Y_A$ and $Y_A^T H Y_R$ will be zero at each subsequent step of the algorithm. Either the algorithm detects that the matrix $Y_R^T H Y_R$ is copositive, or a constraint corresponding to a positive diagonal element of $Y_R^T H Y_R$ is deleted. If the algorithm terminates without having computed a direction of negative curvature, the algorithm has determined that a local minimizer has been found with respect to the constraints that are still present in A_R . However, this conclusion still holds if the deleted constraints are added again, since deletion of constraints may only increase the size of the feasible region. ■

Hence, if Algorithm 6.1 does not terminate at a given step, a constraint with a positive diagonal element of $Y_0^T H Y_0$ is deleted. Recall that Lemma 5.1 implies that the new reduced Hessian is positive definite whenever a constraint corresponding to a positive element of $Y_0^T H Y_0$ is deleted.

The amount of work needed at each step may be reduced by updating Y_0 and $Y_0^T H Y_0$. To show this, we assume that the normal of the constraint $a_i^T x \geq \beta_i$ is deleted from A_0 corresponding to a positive diagonal element of $Y_0^T H Y_0$. Partition A_0 such that

$$A_0 = \begin{pmatrix} A_{01} \\ a_i^T \end{pmatrix}.$$

In order to state the results in compact form, let Λ denote the matrix whose j -th column is λ_j in (3.2) and let Θ denote the matrix whose j -th column equals θ_j . With this partition of A_0 , let the induced partition of Y_0 , Λ and Θ be given by

$$Y_0 = \begin{pmatrix} Y_{01} & y_{0i} \end{pmatrix}, \quad \Lambda = \begin{pmatrix} \Lambda_1 & \lambda_i \end{pmatrix} \quad \text{and} \quad \Theta = \begin{pmatrix} \Theta_{11} & \theta_{i1} \\ \Theta_{1i} & \theta_{ii} \end{pmatrix}.$$

With this partition equation (3.2) may be written in compact form as

$$\begin{pmatrix} H & A_+^T & A_{01}^T & a_i \\ A_+ & 0 & 0 & 0 \\ A_{01} & 0 & 0 & 0 \\ a_i^T & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} Y_{01} & y_{0i} \\ -\Lambda_1 & -\lambda_i \\ -\Theta_{11} & -\theta_{i1} \\ -\Theta_{1i} & -\theta_{ii} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ I & 0 \\ 0 & 1 \end{pmatrix}. \quad (6.1)$$

Algorithm 6.1. *An algorithm for finding a direction of negative curvature*

repeat

 Compute $Y_0^T H Y_0$; Initialize m_A and m_R ;

if ($m_A > 0$) **then**

$k \leftarrow$ argument satisfying $\max_i y_{Ai}^T H y_{Ai}$;

if ($y_{Ak}^T H y_{Ak} > 0$) **then**

 Delete artificial constraint k ; **go to** again; (see Lemma 5.1)

end if;

$k \leftarrow$ argument satisfying $\min_i y_{Ai}^T H y_{Ai}$;

if ($y_{Ak}^T H y_{Ak} < 0$) **then**

$p \leftarrow y_{Ak}$; **go to** exit; (see Lemma 4.3)

end if;

$k, l \leftarrow$ arguments satisfying $\max_{i,j} |y_{Ai}^T H y_{Aj}|$;

if ($y_{Ak}^T H y_{Al} \neq 0$) **then**

$p \leftarrow y_{Ak} \pm y_{Al}$; **go to** exit; (see Lemma 5.2)

else if ($m_R > 0$) **then**

$k, l \leftarrow$ arguments satisfying $\max_{i,j} |y_{Ai}^T H y_{Rj}|$;

if ($y_{Ak}^T H y_{Rl} \neq 0$) **then**

 Compute p ; **go to** exit; (see Lemma 5.3)

end if;

end if;

end if;

if ($m_R = 0$) **or** ($\min_{i,j} y_{Ri}^T H y_{Rj} \geq 0$) **then**

x is a local minimizer; **go to** exit; (see Lemma 4.2)

end if;

$k \leftarrow$ argument satisfying $\min_i y_{Ri}^T H y_{Ri}$;

if ($y_{Rk}^T H y_{Rk} < 0$) **then**

$p \leftarrow y_{Rk}$; **go to** exit; (see Lemma 4.3)

end if;

for $i \leftarrow 1$ **to** m_R **do**

for $j \leftarrow i + 1$ **to** m_R **do**

$negdet \leftarrow (y_{Ri}^T H y_{Ri} y_{Rj}^T H y_{Rj} - y_{Ri}^T H y_{Rj})^2 < 0$;

if ($negdet$) **and** ($y_{Ri}^T H y_{Rj} < 0$) **then**

 Compute p ; **go to** exit; (see Lemma 4.5)

else if ($m_R = 2$) **then**

x is a local minimizer; **go to** exit; (see Lemma 4.4)

end if;

end do;

end do;

$k \leftarrow$ argument satisfying $\max_i y_{Ri}^T H y_{Ri}$;

 Delete regular constraint k ; (see Lemma 5.1)

label again;

until exit occurs;

label exit;

Let \bar{Y}_0 , $\bar{\Lambda}$ and $\bar{\Theta}$ denote the solution of (3.2) in the next step of Algorithm 6.1. Then, \bar{Y}_0 , $\bar{\Lambda}$ and $\bar{\Theta}$ will satisfy the equation

$$\begin{pmatrix} H & A_+^T & A_{01}^T \\ A_+ & 0 & 0 \\ A_{01} & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{Y}_0 \\ -\bar{\Lambda} \\ -\bar{\Theta} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ I \end{pmatrix}.$$

Lemma 6.2. *The quantities $\bar{\Theta}$ and \bar{Y}_0 may be obtained from the solution of (6.1) as*

$$\bar{\Theta} = \Theta_{11} - \frac{\theta_{1i}\theta_{1i}^T}{\theta_{ii}} \quad \text{and} \quad \bar{Y}_0 = Y_{01} - \frac{y_{0i}\theta_{1i}^T}{\theta_{ii}}.$$

Proof. The matrices $\bar{\Theta}$ and \bar{Y}_0 satisfy the equation

$$\begin{pmatrix} H & A_+^T & A_{01}^T & a_i \\ A_+ & 0 & 0 & 0 \\ A_{01} & 0 & 0 & 0 \\ a_i^T & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{Y}_0 \\ -\bar{\Lambda} \\ -\bar{\Theta} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ I \\ a_i^T \bar{Y}_0 \end{pmatrix}. \quad (6.2)$$

Equations (6.1) and (6.2) imply that the barred quantities may be obtained from the equations

$$\bar{Y}_0 = Y_{01} + y_{0i}a_i^T\bar{Y}_0 \quad (6.3a)$$

$$\bar{\Lambda} = \Lambda_1 + \lambda_i a_i^T \bar{Y}_0 \quad (6.3b)$$

$$\bar{\Theta} = \Theta_{11} + \theta_{1i}a_i^T\bar{Y}_0 \quad (6.3c)$$

$$0 = \Theta_{1i} + \theta_{ii}a_i^T\bar{Y}_0. \quad (6.3d)$$

It follows from Lemma 4.1 that $\Theta = Y_0^T H Y_0$. Hence, Θ is a symmetric matrix with $\Theta_{1i} = \theta_{1i}^T$. Equation (6.3d) implies that $\theta_{1i}^T + \theta_{ii}a_i^T\bar{Y}_0 = 0$. The fact that $a_i^T x \geq \beta_i$ is associated with a positive diagonal element of $Y_0^T H Y_0$ implies that θ_{ii} is positive. Substitution in (6.3a) and (6.3c) yields the desired result. ■

Hence, only a rank-one modification of Y_0 and $Y_0^T H Y_0$ is needed at each step of Algorithm 6.1.

Lemma 6.3. *Assume that $y_{0i}^T H y_{0j}$ is zero and $y_{0i}^T H y_{0i}$ is positive at one step of Algorithm 6.1. Also assume that the constraint with normal $A_{0i}^T e_i$ is deleted at this step. At the next step, the column of $Y_0^T H Y_0$ corresponding to the constraint with normal $A_{0i}^T e_j$ is modified only by deletion of the zero element $y_{0i}^T H y_{0j}$.*

Proof. Lemma 6.2 implies that the rank-one modification of column j is zero when $y_{0i}^T H y_{0j}$ is zero. ■

Lemma 6.4. *If, in one step of Algorithm 6.1, it holds that the matrices $Y_R^T H Y_A$ and $Y_A^T H Y_A$ are zero, then they will remain zero.*

Proof. Since both the matrices $Y_R^T H Y_A$ and $Y_A^T H Y_A$ are zero, the only way the algorithm does not terminate is when a regular constraint corresponding to a positive diagonal element of $Y_R^T H Y_R$ is deleted. Lemma 6.3 implies that the matrices $Y_R^T H Y_A$ and $Y_A^T H Y_A$ will remain zero. Only one column of zeros from $Y_R^T H Y_A$ is deleted at each step. ■

Hence, once the matrices $Y_R^T H Y_A$ and $Y_A^T H Y_A$ are zero, they remain zero.

Lemma 6.5. *If $Y_R^T H Y_R$ is positive semidefinite and $Y_R^T H Y_A$ and $Y_A^T H Y_A$ are zero, then Algorithm 6.1 will resolve that x is a local minimizer in at most m_R steps.*

Proof. Lemma 6.4 implies that the matrices $Y_R^T H Y_A$ and $Y_A^T H Y_A$ will remain zero until the algorithm terminates. Hence, the only iteration when the algorithm does not halt is when regular constraints corresponding to positive diagonal elements of $Y_R^T H Y_R$ are deleted. Therefore, at most m_R steps may be taken in the algorithm.

Assume that the algorithm terminates without determining that x is a local minimizer. It follows that a direction of negative curvature must have been computed. But Lemma 6.2 implies that the matrix $\bar{\Theta}$ of the next step is obtained as

$$\bar{\Theta} = \Theta_{11} - \frac{\theta_{1i}\theta_{1i}^T}{\theta_{ii}}.$$

Sylvester's law of inertia implies that $\text{In}(\bar{\Theta}) = \text{In}(\Theta) - \text{In}(\theta_{ii})$. At the initial iteration, Θ is positive semidefinite. The value of the scalar θ_{ii} is positive. Hence, $\bar{\Theta}$ will have no negative eigenvalues. It follows by induction that no direction of negative curvature can be computed. ■

Hence, if $Y_R^T H Y_R$ is positive semidefinite, Algorithm 6.1 determines that x is a local minimizer.

7. Changes in the Working Set

In this section the changes in the working set are described. In the proposed algorithm, either one or two constraints in A will become inactive. In an ICQP method, only one constraint is added or deleted at a time. However, we shall give a scheme that allows deletion of two constraints at a dead point, maintaining the properties of an ICQP method, i.e., the reduced Hessian having at most none nonpositive eigenvalue and the working-set matrix having full row rank.

When a direction of negative curvature, p , is computed, the objective function is strictly decreasing along that direction. The boundedness of φ in the feasible region guarantees that a sufficiently large step along p will violate a constraint. Let a_k denote the normal of the first constraint that is violated. In order to determine how to update A , it is necessary to know if a_k is dependent on the rows of A . The following lemma, given by Gill *et al.* [GMSW88], shows how linear independence may be checked.

Lemma 7.1. *Consider the equations*

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \omega \\ v \end{pmatrix} = \begin{pmatrix} a_k \\ 0 \end{pmatrix}. \quad (7.1)$$

The vector a_k is dependent on the rows of A if and only if the vector ω is zero in the solution of (7.1).

Proof. Suppose that a_k is dependent on the rows of A . In this case, there must exist a vector v such that $a_k = A^T v$, and ω is zero in the solution of (7.1).

Assume that ω is zero in the solution of (7.1). It follows that $a_k = A^T v$, and a_k is dependent on the rows of A . ■

When the algorithm is applied, either one or two constraints leave the working set. The following sections show how to update the working-set matrix.

7.1. One constraint becomes inactive

Assume that p is given by $p = y_{0i}$ and let $a_i^T x \geq \beta_i$ denote the constraint that leaves the working set.

Lemma 7.2. *Assume that p is computed by deleting one constraint from the working set. If a_k is independent of the rows of A , it is added to A , while a_i is maintained in A as an artificial constraint. If a_k is dependent on the rows of A , a_k and a_i are exchanged. In either case, the resulting reduced Hessian is positive definite and working-set matrix has full row rank.*

Proof. If a_k is independent of the rows of A , the new reduced Hessian remains positive definite since only one more constraint is added to the working set. Also, the new working-set matrix has full row rank.

Now assume that a_k is dependent on the rows of A . If a_k and a_i are exchanged, the rows of the new working set will span the same space as the rows of A . Hence, the new reduced Hessian is positive definite. Also, the new working-set matrix has the same number of rows as the old one, and therefore it has full row rank. ■

Hence, after having either added a_k or exchanged a_k and a_i , the new reduced Hessian is positive definite and the new working-set matrix has full row rank.

7.2. Two constraints become inactive

Assume that p is given by $p = \alpha_i y_{0i} + \alpha_j y_{0j}$, where α_i and α_j are both nonzero. Let a_i and a_j denote the normals of the constraints which leave the working set.

Lemma 7.3. *Assume that $a_k = A_2^T v_2 + a_i v_i + a_j v_j$. Then it cannot hold that $v_i = v_j = 0$.*

Proof. Assume that $a_k = A_2^T v_2$. Premultiplication by p^T yields $p^T a_k = 0$. But this could not hold since a_k becomes violated when a sufficiently large step along p is taken. ■

Lemma 7.4. *Assume that p is computed by making two active constraints inactive. If a_k is independent of the rows of A , it is added to A , while a_i and a_j are maintained as artificial constraints. If a_k is dependent on the rows of A and $|v_i| > |v_j|$, a_k and a_i are exchanged. If a_k is dependent on the rows of A and $|v_i| \leq |v_j|$, a_k and a_j are exchanged. In each case, the new reduced Hessian is positive definite and the new working-set matrix has full row rank.*

Proof. Assume that a_k is independent of the rows of A . The new reduced Hessian remains positive definite since only one more constraint is added to the working set. Also, the new working-set matrix has full row rank.

Assume that a_k is dependent on the rows of A . Lemma 7.3 implies that at least one of the scalars v_i and v_j is nonzero. Hence, by performing the specified exchange, the rows of the new working set will span the same space as the rows of A . Hence, the new reduced Hessian will be positive definite. Also, the new working-set matrix has the same number of rows as the old one, and therefore it has full row rank. ■

Hence, after having either added a_k or exchanged either a_k and a_i or a_k and a_j , the new reduced Hessian is positive definite and the working-set matrix has full row rank.

8. Verification of Local Optimality

In this section we describe a complete algorithm for checking if a given dead point x is a local minimizer. In Algorithm 8.1, a direction of negative curvature is found by making one or two constraints leave the working set at a time. If no such direction exists, Algorithm 6.1 yields the result that x is a local minimizer.

If constraints corresponding to positive diagonal elements of $Y_R^T H Y_R$ are deleted in Algorithm 6.1, assumption A2 will no longer hold. In this case, if a direction of negative curvature is computed, the resulting maximum feasible step could be zero and there is a danger of cycling.

However, if Algorithm 8.1 terminates, it will provide either a feasible direction of negative curvature along which a nonzero step may be taken or the information that x is a local minimizer. As shown in Section 6, the algorithm will terminate with the information that x is a local minimizer in the special case when the matrix $Y_R^T H Y_R$ is positive semidefinite.

9. Conclusions

When solving a general quadratic programming problem there may exist certain dead points at which it is very difficult to verify optimality. We emphasize that this difficulty is inherent to the problem, and is independent of the solution method.

Algorithm 8.1. *An algorithm checking for local optimality***repeat**

Apply Algorithm 6.1;

 $local_minimizer \leftarrow$ Algorithm 6.1 implies that x is a local minimizer; **if** ($local_minimizer$) **then** **go to** exit; **else** $a_k \leftarrow$ constraint that is first violated along p ; $\alpha_F \leftarrow$ maximum feasible step along p ; Solve $\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \omega \\ v \end{pmatrix} = \begin{pmatrix} a_k \\ 0 \end{pmatrix}$; (see Lemma 7.1) $indep \leftarrow (\|\omega\| > 0)$; $nr_inactiv \leftarrow$ number of constraints that become inactive; **if** ($nr_inactiv = 1$) **then** $a_i \leftarrow$ constraint that becomes inactive; **if** ($indep$) **then** Add a_k ; **else** Exchange a_k and a_i ; (see Lemma 7.2) **end if**; **else** $a_i, a_j \leftarrow$ constraints that become inactive; **if** ($indep$) **then** Add a_k ; **else** **if** ($|v_i| > |v_j|$) **then** Exchange a_k and a_i ; (see Lemma 7.4) **else** Exchange a_k and a_j ; **end if**; **end if**; **end if**; **if** ($\alpha_F > 0$) **then** $x \leftarrow x + \alpha_F p$; $local_minimizer \leftarrow$ **false**; **go to** exit; **end if**; **end if**;**until** too many iterations;**label** exit:

In this paper, the verification of optimality has been discussed within the context of an inertia-controlling method. We have derived a computational method appropriate for general ICQP methods, that will attempt to determine if a dead point is a local minimizer. The use of artificial constraints may introduce additional dead points. It has been shown that the new procedure does not terminate at such points, unless they are local minimizers.

However, the verification of optimality in the general case is an NP-hard problem, so we would not expect to find a procedure capable of solving a general problem in a reasonable amount of computational effort. In our scheme, there is a potential danger of cycling, and techniques similar to anti-cycling procedures used in linear programming are needed to resolve this problem.

Acknowledgement

We would like to thank Richard Cottle for bibliographical assistance and helpful discussions on the properties of copositive matrices.

References

- [Bor82] J. M. Borwein. Necessary and sufficient conditions for quadratic minimality. *Numer. Funct. Anal. and Optimiz.*, 5, 127-140, 1982.
- [CHL70] R. W. Cottle, G. J. Habetler, and C. E. Lemke. On classes of copositive matrices. *Linear Algebra and its Applications*, 3, 295-310, 1970.
- [Con80] L. B. Contesse. Une caractérisation complète des minima locaux en programmation quadratique. *Numerische Mathematik*, 34, 315-332, 1980.
- [Fle71] R. Fletcher. A general quadratic programming algorithm. *J. Institute of Mathematics and its Applications*, 7, 76-91, 1971.
- [GMSW84] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. *User's guide for SOL/QPSOL (Version 3.2)*. Report SOL 84-6, Department of Operations Research, Stanford University, 1984.
- [GMSW88] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. *Inertia-controlling methods for quadratic programming*. Report SOL 88-3, Department of Operations Research, Stanford University, 1988.
- [Gou85] N. I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Mathematical Programming*, 32, 90-99, 1985.
- [Gou86] N. I. M. Gould. *An algorithm for large scale quadratic programming*. Technical Report CSS 219, Computer Science and Systems Division, AERE Harwell, Oxford, England, 1986.
- [Maj71] A. Majthay. Optimality conditions for quadratic programming. *Mathematical Programming*, 1, 359-365, 1971.
- [Man80] O. L. Mangasarian. Locally unique solutions of quadratic programs, linear and nonlinear complementarity problems. *Mathematical Programming*, 19, 200-212, 1980.
- [MK87] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39, 117-129, 1987.
- [Per72] F. J. Pereira. *On characterizations of copositive matrices*. PhD thesis, Department of Operations Research, Stanford University, 1972.
- [PS88] P. M. Pardalos and G. Schnitger. Checking local optimality in constrained quadratic programming is NP-hard. *Operations Research Letters*, 7, 33-35, 1988.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SOL 89-11	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) On the Identification of Local Minimizers in Inertia-Controlling Methods for Quadratic Programming		5. TYPE OF REPORT & PERIOD COVERED Technical Report
7. AUTHOR(s) Anders L. Forsgren, Philip E. Gill and Walter Murray		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Operations Research - SOL Stanford University Stanford, CA 94305-4022		8. CONTRACT OR GRANT NUMBER(s) N00014-87-K-0142
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research - Dept. of the Navy 800 N. Quincy Street Arlington, VA 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1111MA
		12. REPORT DATE July 1989
		13. NUMBER OF PAGES 21 pages
		14. SECURITY CLASS. (of this report) UNCLASSIFIED
		15. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) This document has been approved for public release and sale; its distribution is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Quadratic programming, local minimizer, NP-hardness.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (see reverse side)		

SOL 89-11: On the Identification of Local Minimizers in Inertia-Controlling Methods for Quadratic Programming, Anders L. Forsgren, Philip E. Gill and Walter Murray (July 1989, 21 pp.).

The verification of a local minimizer of a general (i.e., nonconvex) quadratic program is in general an NP-hard problem. The difficulty concerns the optimality of certain points (which we call *dead points*) at which the first-order necessary conditions for optimality are satisfied, but strict complementarity does not hold.

One important class of methods for solving general quadratic programming problems are called *inertia-controlling quadratic programming (ICQP) methods*. We derive a computational scheme for proceeding at a dead point that is appropriate for a general ICQP method.